



# Robust Ad-hoc Sensor Routing (RASeR) protocol for mobile wireless sensor networks



T. Hayes\*, F.H. Ali\*

Communications Research Group, School of Engineering and Informatics, University of Sussex, UK

## ARTICLE INFO

### Article history:

Received 24 August 2015

Revised 23 June 2016

Accepted 15 July 2016

Available online 18 July 2016

### Keywords:

Mobile

Routing

Network

## ABSTRACT

Robust Ad-hoc Sensor Routing (RASeR) is a novel protocol for data routing in mobile wireless sensor networks (MWSNs). It is designed to cope with the demanding requirements of emerging technologies, which require the reliable and low-latency delivery of packets in highly mobile conditions. RASeR uses blind forwarding, which is facilitated by a novel method of gradient maintenance. The problem of maintaining a gradient field in a changing topology, without flooding, is solved by using a global time division multiple access MAC. Furthermore, it is enhanced with the additional options of a supersede mode, to aid time-critical applications, reverse flooding, to allow sink-to-sensor commands and energy saving sleep cycles to reduce power consumption. Analytical expressions are derived and verified by simulation. RASeR is compared with the state-of-the-art MWSN routing protocols, PHASeR and MACRO, as well as the MANET protocols, AODV and OLSR. The results indicate that RASeR is a high performance protocol, which shows improvements over PHASeR, MACRO, AODV and OLSR. Tested over varying levels of mobility, scalability and traffic, the simulations yield near perfect PDR in many scenarios, as well as a low end-to-end delay, high throughput, low overhead and low energy consumption. The robustness of this protocol and its consistent reliability, low latency and additional features, makes it highly suitable to a wide number of applications. It is specifically applicable to highly mobile situations with a fixed number of nodes and small payloads.

© 2016 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Sensor networks are an important tool for monitoring physical phenomena in the modern world, since they often negate the need for human presence. The nodes' ability to communicate wirelessly removes the need for long wires and enables them to be distributed in an ad-hoc manner wherever and whenever required, which could include harsh and hostile terrains [1]. For this reason, along with the availability of low cost nodes, wireless sensor networks (WSNs) are already commonplace in industry and are becoming more and prevalent in the consumer market. Furthermore, the introduction of mobility to WSNs is an open research issue [2] and can realise the potential for many more emerging applications and will be a key enabling technology in the future of ubiquitous sensing.

One of the main challenges in these mobile WSNs (MWSNs) is the routing protocol, which aims to transport the data generated by the sensors to the sink. This is non-trivial due to the limited

resources, such as bandwidth, power and cost, as well as the movement of nodes. A constantly changing topology means that a fixed path from a sensor to the sink cannot be guaranteed. The more demanding applications also require the consistent delivery of real-time data in highly mobile scenarios, which necessitates a robust, high performance routing protocol.

There are an increasing number of applications for which a MWSN may be utilised, such as wildlife monitoring [3], environment mapping [4] and traffic monitoring in smart cities [5]. There are also applications in emergency scenarios [6] such as the monitoring of vital signs [7] in temporary hospitals and the use of unmanned aerial vehicles (UAVs) in the aiding of search and rescue (SAR) [8].

As such, Robust Ad-hoc Sensor Routing (RASeR) protocol is designed to be a reliable solution, even with the high frequency topology changes of a mobile network. It uses a simple hop-count gradient to allow sensor nodes to blindly forward data towards a single sink. A key issue with this type of routing is in keeping the gradient metric up to date, for this reason RASeR uses a design that combines a global time division multiple access (GTDMA) medium access control (MAC) scheme with the routing protocol. GTDMA simply allows each node in the network to transmit in turn, such

\* Corresponding author.

E-mail addresses: [T.Hayes@sussex.ac.uk](mailto:T.Hayes@sussex.ac.uk) (T. Hayes), [F.H.Ali@sussex.ac.uk](mailto:F.H.Ali@sussex.ac.uk) (F.H. Ali).

that only one node is able to transmit at a time. This eliminated the possibility of collisions and also enables the constant updating of the gradient without the added overhead of flooding or location awareness.

The forwarding technique used inherently takes advantage of route diversity [9], which is designed to utilise multiple paths simultaneously, such that if one route fails there is another still active to deliver the packet. This makes the protocol very dependable in terms of packet delivery and very robust to link failure.

The next section will give an overview of the current state-of-the-art, in terms of general routing for MWSNs, and positions the work. This is followed by a description of the proposed protocol in Section 3, which includes details on how the gradient field is maintained and the mechanism by which nodes forward data. This is followed by a description of additional options that can be incorporated within RASER, which can make it more suitable to a higher number of applications. Section 5 derives analytical expressions and Section 6 presents simulation results, which measure packet delivery ratio, end-to-end delay, overhead, throughput and energy consumption in various scenarios. Finally, Section 7 concludes the paper.

## 2. Literature review

MWSN routing protocols generally take influence from both WSN and mobile ad hoc network (MANET) routing protocols, which all share common limitations, such as bandwidth, power and cost. WSNs often share the same aim as MWSNs, in that they wish to route data from many sensors to a single sink. However, WSNs are normally considered to be static and so the associated routing protocols are often unable to cope in a mobile scenario [10]. Alternatively, MANET protocols are designed to be able to cope with the mobility of nodes, however they aim to allow end-to-end communication to occur between any two nodes [2]. This extra functionality is often not required by MWSNs and so the additional overhead is unnecessary. Combined with the high packet delivery ratios and low delays that are demanded by emerging applications, the ideal routing solution for a MWSN is one that can handle the mobility of nodes and allows data to be forwarded from the sensors to the sink in a reliable and timely manner. This set of requirements make the problem of routing in a MWSN a unique challenge, which will require new specifically designed solutions. For this reason there have been many routing protocols designed for MWSNs. As such, this section will give an overview of the current literature, which highlights the different techniques and commonly used protocols in MWSN routing.

Protocols designed for MANETs are usually defined as either proactive or reactive. Proactive protocols, attempt to maintain an active route from every node to every other node. This is often done with flooding, which can cause significant congestion in large networks. Alternatively, the reactive style of routing instructs nodes to discover routes only when they are needed. This can often reduce the amount of control overhead, which makes protocols such as AODV (Ad-hoc On-demand Distance Vector) [11], DSR (Dynamic Source Routing) [12] and Ex-OR (Extremely Opportunistic Routing) [13] a more common choice for mobile networks.

This can easily be seen in the number of reactive MANET protocols that have been adapted for use in MWSNs. For example, both AODV++ [14] and AODV-PSR (AODV with Pre-emptive Self Repair) [15] have been adapted from AODV. AODV++ uses a technique of route choice, which is influenced by link reliability, energy levels and traffic rates. Whereas, AODV-PSR attempts to predict link breaks and find an alternative.

Another example is OR-RSSI (Opportunistic Routing-RSSI) [16], which is influenced by Ex-OR. In OR-RSSI the sink emits a high power beacon, the sensor nodes then use the RSSI of the beacon

to give a measure of distance from the sink. This distance measure is combined with the node's mobility to give an opportunistic probability.

Alternatively, GOR (Geographically Opportunistic Routing) [17] is another opportunistic routing protocol, which uses location information to forward data towards the sink. It splits up the network into sections and each node tries to forward its data to another node that's in a section closer to the sink. Firstly nodes will opportunistically attempt to transmit to the furthest section possible, before trying increasingly closer sections.

ADSR (Angle-based DSR) [18] has been adapted from DSR and also uses location information. The geographic information is used at each node to determine the angle between itself, the fixed sink and a potential forwarding node. This information is used to ensure that the data is always transmitted towards the sink in a greedy manner.

Some other techniques have been used to improve the reliability of packet delivery, such as RRP (Robust cooperative Routing Protocol) [19]. RRP uses node's cooperation to mitigate some of the effects of a changing topology by allowing a node that has overheard a failed transmission to retransmit the packet if it is within range of the intended recipient.

In contrast to MANETs, WSN routing protocols are commonly categorised by their hierarchical or flat structure. Hierarchical protocols, such as LEACH (Low Energy Adaptive Clustering Hierarchy) [20], split the nodes up into clusters based on their locality to each other. Each cluster then elects a cluster head, which receives data from the other nodes in that cluster and then forwards it on to the sink. This technique has been shown to reduce energy consumption on static sensor networks. However, in a mobile network, the frequent movement of nodes from one cluster to another can cause large amounts of overhead in terms of becoming associated with different cluster heads.

Some hierarchical protocols have been adapted for MWSNs, such as LEACH-M (LEACH-Mobile) [21], which is based on LEACH and allow nodes to dynamically switch between clusters. This is done by providing a mechanism of determining when a node has become disconnected and should then join another cluster. LEACH-ME (LEACH-M Enhanced) [22] is an improvement on this, which selects cluster heads based on each node's mobility. This means that the nodes with the lowest mobility become cluster heads, making the network more stable.

Similarly, CBR Mobile-WSN (Cluster Based Routing for Mobile WSNs) [23] is a hierarchical protocol, which allocates empty transmission slots to nodes without a cluster. So, if a node becomes disconnected it may still transmit to a cluster head before it becomes associated with a new cluster, which reduces packet loss. Additionally, ECBR-MWSN (Enhanced Cluster Based Routing for MWSNs) [24] improves upon CBR Mobile-WSN by selecting cluster heads that are closest to the sink and have the most residual energy, which helps to prolong the lifetime of the network. Also, MBC (Mobility Based Clustering) [25], generates a suitability metric from measures of estimated connection time, residual energy, the cluster heads node degree and distance. This metric is then used to make a more informed decision on cluster head selection.

Another hierarchical protocol is ZBR (Zone Based Routing) [26], which assumes that each node has knowledge of its own location. Each node then knows which cluster it is associated with by its geographic location, unlike LEACH which selects cluster heads first and then defines cluster members by those that are within radio range. Also, ZBR selects cluster heads by having each node broadcast a mobility factor and then the least mobile node is selected. LFCP-MWSN (Location aware Fault-tolerant Clustering Protocol for MWSNs) [27] also uses location awareness to define clusters but determines cluster heads using both mobility and energy levels. This helps to improve the stability and lifetime of the network.

In contrast to the hierarchical structure, flat protocols, such as SPIN (Sensor Protocol for Information via Negotiation) [28] and Directed Diffusion (DD) [29] require no infrastructure, which makes them preferable in mobile networks. DCBM (Data Centric Braided Multipath) [30] was adapted from the query orientated approach used in DD to create a multipath routing protocol for MWSNs. The initial query is flooded through the network with the intermittent nodes recording the path. This information is then used to determine multiple paths back to the sink once the appropriate node has received the query.

The recent protocol, MACRO (Mobility Adaptive Cross-layer Routing) [31], presents a state-of-the-art cross-layer approach to routing in MWSNs. It exploits the interaction between the layers by sharing information such as average speed and RSSI data. Similarly to AODV, MACRO discovers routes using the common route request/route reply technique, however it limits the flooding by minimizing the number of nodes that forward the requests. In order to improve the reliability of packet delivery, routes are given an expiry time based on the mobility of the node and the link quality. This ensures that only the most reliable routes are used. Energy is saved by using an adaptive duty-cycle in the MAC layer and also adjusting the transmission power of the node's radio.

In general, from the existing protocols surveyed, it is suggested that whilst protocols designed for mobile networks often incur a higher level of overhead, the best performance tends to be from those that keep this to a minimum. This makes gradient routing a particularly applicable technique to the scenario of MWSNs. The static WSN protocol GBR (Gradient Based Routing) [32] initially establishes a hop-count metric at each node, such that data can be forwarded down the gradient to the sink. This is done in a setup phase by flooding the network, and then nodes share their hop-counts with their neighbours so nodes can choose the most appropriate next hop. In MWSNs the movement of nodes would require the network to be flooded periodically to keep the gradient updated, which would cause significant network congestion.

The issue of maintaining an up-to-date gradient has also been tackled with the use of location information in GPSR (Greedy Perimeter Stateless Routing) [33]. GPSR works in a similar way to GBR but with a geographic gradient, such that nodes greedily choose a forwarding neighbour that is physically closest to the sink. One drawback to the use of location awareness is in the reliability and power consumption required to obtain the geographic information. Another issue is the dead-end problem [34], where a node is locally maximal and so has no neighbours that are closer to the sink meaning the packet has reached a dead-end. In GPSR this is addressed with the use of the right-hand rule. The PAGER (Partial-Partition Avoiding Geographic Routing) [35] algorithm was proposed as an alternative solution to the dead-end problem, which identifies nodes that are likely to forward data to a dead-end node and provides them with an alternative route. PAGER-M (PAGER-Mobile) [35] uses this algorithm specifically in the context of MWSNs. GPSR-MS (GPSR with Mobile Sensors) [36] also adapts GPSR for MWSNs by using a different gradient metric, which takes into account the forwarding node's distance from the sink as well as its speed and direction.

In static WSNs, DBO (Directed Broadcast with Overhearing) [37] is one example of a protocol that uses gradient routing but with the technique of blind forwarding. Blind forwarding is a method of forwarding data in a gradient based routing scheme, in which a transmitting node will not select a single forwarding neighbour, but blindly broadcast its packet to all neighbours within range. Each of the receiving nodes can then compare their own gradient to that in the received packet and decide whether they should forward the packet or not.

This style of routing is somewhat opportunistic in its nature, however in opportunistic routing subsets of forwarding nodes are

defined, either at the source or on a hop-by-hop basis. Then the forwarding node is selected through a coordination process usually based around an acknowledgement (ACK) scheme or request-to-send/clear-to-send (RTS/CTS) handshake [38]. Contrastingly, blind forwarding considers all nodes to be potential forwarding nodes so there is no forwarding node determination. This means that the technique cannot use any forwarding node coordination and whilst the protocol may save some overhead from eliminating the use of ACKs and RTS/CTS messages, depending on the MAC layer being used, this can potentially reduce the reliability given from the use of ACKs and more collisions may occur by removing the RTS/CTS mechanism.

In MWSNs, PHASeR (Proactive Highly Ambulatory Sensor Routing) [39] takes advantage of the blind forwarding technique, but the addition of mobility means that a method of gradient maintenance must be employed. For this the protocol uses a GTDMA MAC, which allows a hop-count to be kept up-to-date. The issues with this protocol are in the fact that it is designed for applications in which nodes are generating data with a fixed period. This inspired the use of encapsulation in the packet structure, such that a packet could contain data from multiple nodes. This means that, in every timeslot the appropriate node will transmit a packet containing as much new data as possible. These large packet sizes require longer timeslots, which can cause long delays and also reduce the frequency with which the network can update the gradient metric. As nodes have to wait for a cycle before being able to transmit, combined with the fact that the packet size has a finite limit, it can be the case that some data has to be dropped. To aid with this, PHASeR uses a queuing algorithm, which simply overwrites old data that's received from the same node, such that it prioritises newer data over older data.

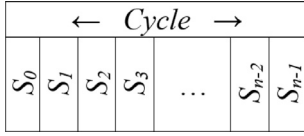
RASeR is a novel, high performance routing protocol, which is intended to be widely applicable to many next generation MWSN scenarios. Its design promotes the low latency delivery of data with minimal packet loss. Many of these advantages come from addressing the limitations of PHASeR.

Similarly to PHASeR, RASeR also uses the technique of blind forwarding in a MWSN to forward packets towards the sink, and also takes advantage of the novel application of a GTDMA MAC in order to handle the issue of gradient maintenance in a changing topology without having to flood the network. However, instead of using data encapsulation, packets are limited to a single piece of data. This greatly changes the structure, implementation and further design choices of the protocol. The advantage is in the reduction of the length of timeslots, meaning that nodes have to wait a much shorter time before being able to transmit. The consequences of this are in the fact that data can be delivered much faster and the protocol is able to cope with even the highest levels of mobility. RASeR may also implement a first-come-first-serve style of queuing and reduce packet loss. This approach is low in overhead and works well even in the most demanding of environments, which makes it highly suitable to MWSNs and applicable to a very wide variety of applications.

A detailed description of every aspect of the protocol is given in the next section, which highlights the protocol's unique construction.

### 3. Protocol description

The envisaged scenario for this work is in MWSNs, in which it is assumed that the nodes are provided with some form of mobility platform. This maybe a robot or an autonomous vehicle on land, in the air or underwater. Given the expense of these autonomous platforms, a relatively small number of privately owned nodes are anticipated to be used and reused for multiple missions. Since individual networks of this kind are most likely owned by



**Fig. 1.** GTDMA cycle structure, showing how a cycle is made up of  $n$  time slots. Slot  $S_0$  will be assigned to the sink,  $S_1$  will be assigned to the first sensor node,  $S_2$  will be assigned to the second sensor node.

institutions or individuals and deployed for specific purposes, unique to their owners, they are unlikely to require other foreign nodes joining the network during a mission. Also, as these nodes are not disposable, they are required to be deployed with enough power to return to the base when the mission is over. This means that no nodes are expected to leave the network, except in the event of node failure. Given these expectations, the number of nodes is anticipated to remain fixed for the duration of the operation, though it is also noted that this may make the protocol less appropriate for some applications. However, this is acceptable for the sake of creating an optimised high performance protocol.

### 3.1. Global TDMA

RASer allows each node to transmit one at a time in a pre-defined order; in other words, each node is assigned a single time slot, which is large enough to transmit a single packet. The order in which the node's time slots occur is fixed and loops cyclically. As such, a cycle is the length of time it takes for each node to transmit and a slot is the time it takes to transmit a single packet. This is illustrated in Fig. 1, which shows how each cycle is made up of  $n$  time slots, where  $n$  is the number of nodes in the network.

GTDMA is deterministic and entirely contention free, meaning that no collisions can occur, which reduces the chances of packet loss. Since the time slot lengths and node numbers are set before deployment, the protocol can be uniquely optimised for each new mission, making RASer highly adaptable. The protocol is also scalable whilst maintaining a highly reliable level of packet delivery, though increasing the number of nodes results in an increased level of delay.

The choice of using a GTDMA is contrary to the traditional use of a dynamic TDMA, in which time slots are allocated dynamically by a centralised authority. This is because, the setup phase used to allocate slots, requires additional overhead and in a highly mobile environment, will need to be run regularly. Although, one of the main advantages to the classic TDMA is that it allows nodes to leave and join the network and can also take advantage of unused bandwidth. However, given the assumption of a fixed number of nodes, as explained above, it is unlikely that nodes will be required to leave or join the network. This having been said, the main motivation for using a GTDMA, is to facilitate the constant maintenance of the gradient field. By using the GTDMA MAC it is assured that each node will broadcast in a strict order in a deterministic manner, which also allows for the gradient to be refreshed with the highest possible frequency. Additionally, the MAC guarantees that when a node is allowed to transmit it has had the opportunity to overhear all the other nodes within its transmission range, which will enable it to calculate its gradient with a high level of precision. Contrastingly, another MAC, such as CSMA/CA, could be used with timers to indicate when each node should broadcast its topology information. However, this doesn't guarantee that a node will be transmitting data with the most up-to-date gradient metric. By utilising GTDMA in this way the routing protocol is reliant on this particular MAC layer and as such they can be implemented together, symbiotically, or alternatively on an existing system that

already uses a GTDMA scheme, so long as sufficient communication between the two is available.

As a trade-off for the reliability of the GTDMA MAC, one of the primary concerns is the latency that nodes will experience by having to wait for their allocated time slots before they can transmit. However, packet end-to-end delay time is also kept low by the fact that there is no forwarding node selection, no collision avoidance mechanism and no retransmissions. Additionally, in cases where only small packet sizes are required, the cycle time will be kept low and subsequently the packet latency will also be low. The GTDMA scheme also yields high levels of reliability in comparison to contention-based MACs, which is a major advantage.

The use of GTDMA will also require synchronisation, for which there are a number of available methods as surveyed in [40]. One of which is RBS (Reference Broadcast Synchronisation) [41], which uses a third party to broadcast a reference beacon. The protocol assumes that all nodes will receive the beacon simultaneously, subsequently, nodes will synchronise based on the arrival time of the reference beacon. Within the GTDMA context, the sinks timeslot could be used to broadcast a high powered network-wide reference beacon. Another method maybe to use the deterministic nature of the GTDMA MAC. Since every node knows when each timeslot is expected to start, when a node hears another node's broadcast it compares the arrival time with the expected slot start time. Additionally, if a localisation technique like GPS is used to retrieve positional information, the transmitted clock times can also be utilised for synchronisation [42].

Another key consideration with the use of TDMA based schemes is that of external interference. There are many techniques that can be used to reduce or even eliminate the effects of interference. One of the best methods is the use of sophisticated radios that employ spread spectrum technology with frequency hopping or spreading codes. Additionally, the popular 802.11 radios are able to transmit on a number of channels and one could be selected that has little or no other traffic. Furthermore, depending on the application, the network may be deployed in sparsely populated areas such as jungles, desserts, in space or at sea, or by institutions that have their own frequency band, such as the military or the coast guard. It should also be noted that in this particular protocol the use of multipath routing will also help alleviate the effects of external interference.

### 3.2. Hop count determination

The hop count is a simple metric, which indicates a node's distance in hops from the sink. Local topology information is shared to maintain this hop count at each node. The hop-counts are then used to ensure data is always forwarded towards the sink.

Even though the nodes will often require location awareness for reporting positions to the sink, it is still preferable to use a hop count metric for routing. This is because the location information is not considerate of the topology, which can cause issues such as the dead-end problem.

Using the GTDMA MAC, each node has the opportunity to transmit once in a cycle. In every slot nodes are required to transmit a data packet, as in Fig. 2, or a beacon packet if the node has no data to forward. A beacon packet is simply the first two fields of the data packet, namely the node ID and hop count. In this way, each node will have the chance to overhear the transmission of every node that's in range, before its own timeslot comes around. In other words a node determines its position in the network using only the topology information, which is the hop count in this case, that is locally available to it from its one-hop neighbours.

This means that in a single cycle, one node will hear all of its neighbours transmit at least a beacon packet. By overhearing these transmissions a node can determine its own hop count, which is



Field name	Node ID	Hop Count	Data	Priority bit	Packet ID
Size (bits)	$\lceil \log_2 n \rceil$	$\lceil \log_2 n \rceil$	$L_{data}$	1	$\lceil \log_2 n \rceil$
Total size (bits)	$L_P = 3 \cdot \lceil \log_2 n \rceil + L_{data} + 1$				

**Fig. 2.** RASer packet structure, where  $n$  is the number of nodes,  $L_{data}$  is the size of data,  $L_P$  is the total size of a data packet and  $\lceil \cdot \rceil$  is the ceiling function.

simply the lowest hop count of its neighbours plus one. For example, Fig. 3 shows a sink and 8 sensor nodes where all but one is labelled with its hop-count from the sink. Initially node 'A' has just entered the network and has an unknown hop-count. The GT-DMA structure means that each node will transmit in turn so after listening to the broadcasts of its neighbours, node 'A' will have heard the hop-counts 3, 4, 3, 2, and 1. From these values node 'A' can take the lowest value and add one to determine its own hop count of 2, as shown in the second part of the figure. The third part shows how, on the next cycle, the node which originally had a hop-count of 4 has updated the value to 3, which reflects the new state of the network.

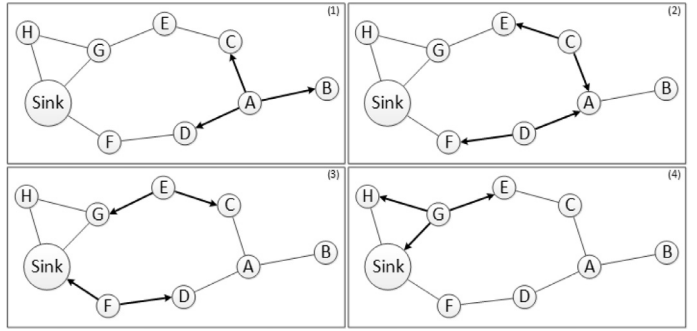
Additionally, node 'A' could also be considered to have moved from another part of the network to its current location. In this case it may initially have any hop count. Then after listening to one cycle of its neighbours broadcasts it can determine its new hop count just as previously described.

It should be highlighted that the novel use of this approach allows nodes to regularly determine their hop-count by taking advantage of the cyclic nature of the GTDMA MAC. The fact that each node is allowed to overhear the broadcasts of every other node in the network before having to transmit is facilitated by the MAC layer and enables the updating of hop-counts. Consequently the issue of gradient maintenance is mitigated and, by only sharing local topology information, flooding the network is not required.

### 3.3. Forwarding data

RASer uses the blind forwarding technique to forward data along a gradient towards the sink, so the decision to forward data is made at the receiving node on a hop by hop basis. In other words, when a node transmits, its broadcast is overheard by all of its neighbours. Each neighbour can then compare the hop count contained in the received packet with its own. Subsequently, if the node's hop count is less than the received hop count, then the packet should be forwarded. If the node's hop count is greater than the received hop count, then the packet should be dropped. Alternatively, if the node's hop count and the received hop-count are equal the packets *status* should be taken into account.

The priorities are used to control the number of routes a packet may take, in this way the redundancy can be kept to a minimum, whilst still increasing the protocol's reliability. Each packet has a



**Fig. 4.** Forwarding data with priorities; (1) Data is generated at node A and broadcast to its neighbours. (2) Based on the hop count gradient, only nodes C and D forward the packet. (3) Again due to the gradient, only nodes E and F forward the data. (4) Here, due to the use of priorities, only node G forwards the data.

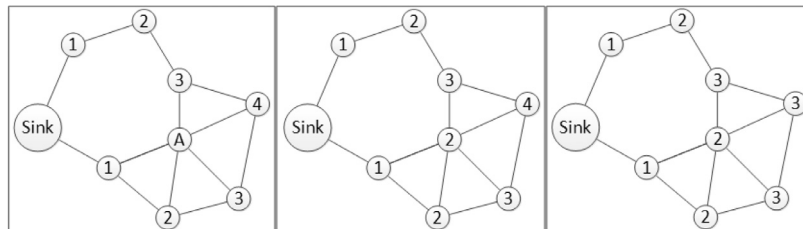
priority bit, which designates its status as either *priority status* or *diversity status*. The *status* of a packet is indicated by the state of its priority bit, which differentiates between *priority packets* and *diversity packets*. When a node receives a packet it stores it in a queue, so before a node's time slot it must decide which packet to transmit; packets with priority status are given precedence over those with diversity status.

The use of diversity packets is designed to increase the route diversity of the protocol without impeding the delay times of the priority packets. So, the diversity packets will increase the number of paths a piece of data may take to the sink, but priority packets will always be transmitted first. Based on this, the oldest priority packets in the queue will be transmitted first, followed by the diversity packets; this is known as *normal mode*.

The determination of priorities is as follows:

- If the node's hop-count is lower than that in the received packet, then the packet should be forwarded with the same priority as it was received.
- If the node's hop-count is higher than that in the received packet, then the packet should be dropped regardless of its priority.
- If the node's hop-count is the same as that in the received packet and the packet has priority status, then the packet should be forwarded with diversity status.
- If the node's hop-count is the same as that in the received packet and the packet has diversity status, then the packet should be dropped.

This use of priorities can be seen in Fig. 4, in which the eight nodes are labelled from A to H. In the first part of the figure, node A generates a packet and broadcasts it to its neighbours with a hop-count of 3 and priority status. Node B has a hop-count of 4 and as such drops the packet. Node D has a hop-count of 2 and stores the packet for forwarding with priority status. Node C has a hop-count of 3, which is the same as node A, so it stores the packet with diversity status. In the second part of the figure, node



**Fig. 3.** Hop-count determination, in which node 'A' has just arrived with an unknown hop-count. Then, from the broadcasts of its neighbours it derives its own hop-count and the nodes around it subsequently adjust their own hop-counts as well.

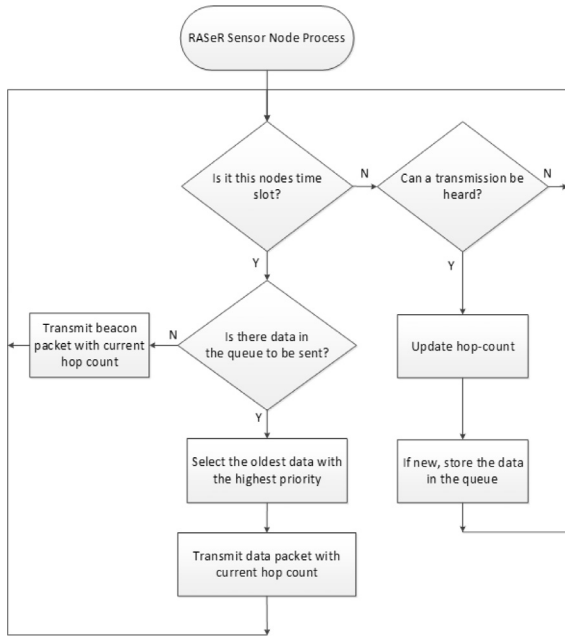


Fig. 5. Flow chart overview of RASer protocol in normal mode.

*D* is shown to broadcast the packet, which will be dropped by node *A* and stored with priority status by node *F*. The third part of the figure shows how node *F* will also broadcast the packet, which is subsequently received by the sink. On the other path, node *C* will broadcast the packet with diversity status, which will be dropped by node *A* and stored for forwarding by node *E*. Node *E* will then broadcast the packet with diversity status, which will be dropped by node *C* and stored by node *G*. The final part of the figure illustrates node *G* broadcasting the packet with diversity status, which will be dropped by node *E* and received by the sink. Node *H* will also hear the transmission from node *G*, however as both nodes have the same hop-count and the packet has diversity status, node *H* will drop the packet.

### 3.4. Normal mode of operation

The general operation of RASer is shown by the flow chart depicted in Fig. 5, which shows the basic algorithm used at each time slot. So initially, at the beginning of a new time slot, nodes must determine whether it is their dedicated slot within the GTDMA cycle. This is done by comparing the node's ID with the time slot number, whereby a node's ID dictates its time slot number. If it is not a node's time slot, then it will listen for a transmission. If the node hears a transmission, then it will first update its hop-count if necessary, and then store any new data in a queue. Else, if there is no transmission heard, then the node may sleep for the remainder of the timeslot. If it is a node's turn to transmit, it will first check whether it has any pending packets to be sent in its queue. If there is data to be sent, the oldest data with the highest priority will be selected using a simple selection algorithm. If there is no data to transmit, the node will broadcast a beacon packet. The selected packet is then broadcast to all of the node's neighbours and the process is repeated. The selection algorithm has a worst case time complexity of  $O(q)$ , where  $q$  is the queue length, this low complexity will also help to reduce power consumption by decreasing processing delay times.

This figure also highlights how the routing tasks performed are based on the GTDMA schedule, which is a key element of the protocol.

The key characteristics of RASer, which make it so robust, are its use of cooperative diversity, this enables any node to become a relay and in turn creates route diversity. Route diversity allows a packet to travel along more than one path to the sink simultaneously, which significantly reduces the detrimental effect of frequent topology change. In addition, the use of a flat network structure means that every node behaves in the same way, which allows the protocol to be kept simple and can easily be deployed on large numbers of nodes.

## 4. Additional options

The variety of applications that a MWSN could be used for will all have very different needs and requirements. For this reason, three additions to the protocol are outlined here. These are optional modes that can be implemented if necessary depending on the application.

### 4.1. Supersede mode

In some applications it may be advantageous for the sink to receive the latest data rather than every acquired sample. This would be the situation if newly generated data would cause the previous samples to become redundant. For example in the application of UAV aided SAR, where the target is out at sea and being carried by the current. In this case it's more important to report the last known location of the target than its previous whereabouts since the old location is redundant. In this type of situation RASer can be set to work in *supersede mode*, in which only the newest packets from each node are kept in the queue. In this way, packets that are considered out-of-date are dropped, which allows the newer packets to traverse the network faster. A node will consider a packet to be out-of-date if it receives another packet from the same source node that has a more recent packet ID. The new packet will then replace the old one. This mode will incur higher packet loss than normal mode, but the lost packets will be the old, irrelevant data. The added packet loss will improve the end-to-end delay of the important data.

### 4.2. Reverse flooding

The majority of sensor network protocols only consider the flow of data from the sensor nodes to a sink, however in most applications it is necessary to disseminate some messages in the opposite direction. For this reason, RASer has a built in reverse flooding mechanism, which can be used to transmit a message from the sink to every node in the network. The technique simply requires each node to retransmit the sink's message once. The packets from the sink are the same as the one shown in Fig. 2, but they always have a node ID of zero and a hop count of zero.

Additionally, the reverse flooding mechanism may also be used to implement a hierarchical based synchronisation protocol such as TPSN (Timing-Sync Protocol for Sensor Networks) [43].

### 4.3. Energy saving

Due to the fact that nodes are generally battery powered, they have a finite amount of energy and as such one key challenge in most sensor networks is that of energy consumption. So, in order to extend the lifetime of the network, nodes need to use as little energy as possible. In RASer it is possible to schedule network-wide sleep cycles, such that no nodes will transmit during this time and the entire network can essentially sleep for a defined amount of time. A *round* is made up of a number of active cycles followed by a number of sleep cycles, both of which can be defined individually, such that any fraction of sleep time to active time can

be realised. In general, much of RASer's energy consumption comes from the broadcasting of beacon packets in order to cope with a highly dynamic topology. However, if the topology is not moving so fast, then the number of beacon packets needed to maintain the gradient field is reduced and sleep cycles can be used to save energy with minimal effect on the protocol's reliability, though it will increase the end-to-end latency.

In terms of determining a suitable ratio of sleep slots to active slots, the measure of average link lifetime can be used to determine how rapidly the topology is changing. This gives a minimum service time, since the network needs at least a full cycle to update the gradient. It is also important that the data is also given time to be delivered successfully, so the packet generation rate of the network needs to be taken into account. Furthermore, the delay requirements of the application will also need to be considered, since there is a direct relationship between sleep cycles and packet delivery latency. The best way to evaluate the ideal sleep cycle characteristics for an application would be to use the analytical expressions given in the next section.

In some cases it may also be advantageous to shut down other parts of the node, such as the sensors or mobility, during these cycles, in order to further reduce the energy consumption.

It should also be noted that energy can also be saved by extending the length of each timeslot; since only a single node will make a single transmission during one timeslot, the remainder of the slot duration can be spent sleeping.

## 5. Mathematical analysis

In this section analytical expressions are derived for packet delivery ratio, average end-to-end delay, throughput, overhead and energy consumption, and are intended to characterise the performance of RASer. The expressions focus on the analysis of the protocol only, so physical layer errors are not considered, however a channel model could be included in future work.

### 5.1. Packet delivery ratio

Packet delivery ratio (PDR) is one of the key metrics in any routing scheme as it is a good measure of the quality of a protocol. It is defined as the fraction of packets successfully received,  $P_{rx}$ , out of the all of the packets created,  $P_{tx}$ , and is given as:

$$PDR = \frac{P_{rx}}{P_{tx}}. \quad (1)$$

Since the GTDMA MAC layer used with RASer is contention free there will be no packet loss from packet collisions. For the sake of developing analytical expressions it is assumed that the network is well connected and as such nodes will not become disconnected. As such the primary cause of packet loss will be through link breakages on the path of a packet.

Taking an expression for the average link lifetime,  $t_{av}$ , adapted from [44]:

$$t_{av} = \frac{d_{link}}{\bar{v}} = \frac{4 \cdot r}{\pi \cdot v_{max}} \quad (2)$$

where  $d_{link}$  is the link distance,  $\bar{v}$  is the relative velocity between the transmitter and receiver,  $v_{max}$  is the maximum speed that a node is capable of and  $r$  is the transmission radius of the nodes. So, during the time between a packet is created and received at the sink, the probability of a link breaking,  $P_{break}$ , is

$$P_{break} = 1 - \left(1 - \frac{\pi \cdot v_{max}}{4 \cdot r}\right)^{D_{av}} \quad (3)$$

where  $D_{av}$  is the average end-to-end delay of a packet. Then the expected number of broken links,  $L_{broken}$ , is given by multiplying

the expected number of links by the probability of a link breaking. The expected number of links in the network is derived through the multiplication of the probability that two nodes are within communication range with all of the possible two node combinations.

$$L_{broken} = P_{break} \cdot \binom{n}{2} \cdot \left(\frac{\pi \cdot r^2}{L^2}\right) \quad (4)$$

$L$  is the length of the square network and  $\binom{n}{2}$  is the binomial coefficient of "n choose 2". Assuming that a link break on the path of a packet will cause packet loss, the packet loss ratio (PLR) is given as:

$$PLR = 1 - (1 - P_{break})^h \quad (5)$$

where  $h$  is the average number of hops between the source node and the sink. This leads to PDR being calculated as

$$PDR = 1 - PLR = \left(1 - \frac{1}{t_{av}}\right)^{D_{av} \cdot h} \quad (6)$$

The average hop-count of a node from the sink is taken from [45] and given as:

$$h = \frac{d_{av}}{d_{hop}} = \frac{2 \cdot L}{3 \cdot r \cdot \cos\left(\frac{\pi}{2 \cdot N_n}\right)} \quad (7)$$

where  $d_{av}$  is the average Euclidean distance between the source and destination,  $d_{hop}$  is the average distance of a single hop and  $N_n$  is the expected number of neighbours to each node. The equation is based on the assumption that the node's neighbours are distributed evenly around it. From this, the expected distance, relative to the sink, is used to estimate how many hops are required.  $N_n$  is given by

$$N_n = \left(\frac{\pi \cdot r^2}{L^2}\right) \cdot (n - 1). \quad (8)$$

This gives a final PDR expression as

$$PDR = 1 - \left(\frac{D_{av} \cdot \pi \cdot v_{max} \cdot h}{4 \cdot r}\right). \quad (9)$$

This expression highlights the fact that increasing the speed of the nodes or the distance to the sink will increase the packet loss. It also has a dependency on the end-to-delay, which will be derived in the next section.

### 5.2. Average end-to-end delay

End-to-end delay is the time taken between a node generating a packet and that packet being received by the sink. It can generally be considered to be the delay at each node,  $T_q$ , multiplied by the number of hops:

$$D_{av} = h \cdot T_q. \quad (10)$$

It is first assumed that the arrival rate of traffic to each node,  $\lambda$ , follows a Poisson distribution and each node is considered to be a single server. Due to the deterministic nature of the GTDMA MAC, the service time,  $T_s$ , is constant so each node can be modelled as an M/D/1 first come first serve (FCFS) queue. We base the node's packet arrival rate on the packet creation rate,  $f_p$ , multiplied by the approximate number of nodes whose data will be forwarded:

$$\lambda = \frac{f_p \cdot (n - 1)}{N_n^2} + f_{psink} \quad (11)$$

where  $f_{psink}$  is the packet generation rate of the sink. This equation takes the network-wide packet generation rate and restricts it by an approximation of the number of nodes that are expected to

forward data through a single node. Since all sink packets will be received these are added in their entirety.

As each node gets the chance to transmit once in a GTDMA cycle, the service time is

$$T_s = \frac{\Delta \cdot n(S_A + S_S)}{S_A} \quad (12)$$

where  $S_A$  is the number of active slots in the round,  $S_S$  is the number of sleep slots in the round and  $\Delta$  is the length of a single time slot, which is given by:

$$\Delta = \frac{L_p}{R_b} + \frac{r}{c} \quad (13)$$

where  $R_b$  is the bit rate and  $c$  is the propagation velocity of the signal. In real world implementation additional hardware specific parameters will also need to be taken into account when determining slot length. These include the radios transmit/receive switching time, sending data along an internal bus as well as processing time. Also, to reduce the effects of clock drift, the timeslots can be enlarged, which will help delay the time it takes for the nodes to become too far out of sync.

Using the Pollaczek–Khinchin formula for the mean time in the system,  $T_q$ , the delay can be described as:

$$D_{av} = \frac{h \cdot T_s \cdot (2 - \lambda T_s)}{2 \cdot (1 - \lambda T_s)} \quad (14)$$

As expected the delay is a function of the number of hops, so the further a node is from the sink the longer the end-to-end delay. Also, the time taken at each node is dependent on the arrival rate of packets and the time it takes to service each one. This is as anticipated since the higher the arrival rate, the longer the queue will be and thus the delay will increase.

### 5.3. Throughput

In this work, throughput, TP, is defined as the number of data bits successfully delivered to the sink, per second, over the entire simulation time. This yields the expression

$$TP = \frac{L_{data} \cdot N_p \cdot PDR}{T_t} \quad (15)$$

where  $N_p$  is the total number of packets produced and  $T_t$  is the total deployment time of the network. The expression shows how the number of packets produced has a large effect on the throughput, as does the number of lost packets.

### 5.4. Overhead

Overhead, OH, is an important aspect in routing protocols as large amounts of overhead can create delays and, in worst case scenarios, cause packet loss. Generally there are two types of overhead; control overhead and packet overhead. Control overhead is the fraction of bits in control packets over bits in data packets. Packet overhead is the fraction of bits in each sensor data packet that is not data.

In this paper, both packet and control overhead are taken into account and measured relative to the quantity of data bits that are received by the sink. In other words, the overall overhead is characterised as the total number of bits transmitted per successfully delivered data bit:

$$OH = \frac{B_{tx}}{L_{data} \cdot N_p \cdot PDR} \quad (16)$$

where  $B_{tx}$  is the total number of bits transmitted.

$B_{tx}$  is given by

$$B_{tx} = \left[ \left( \frac{S_A}{S_A + S_S} \right) \cdot \left( \frac{T_t}{\Delta n} \right) - (N_f \cdot f_p \cdot T_d) \right] \cdot L_{p_{beacon}} \cdot n + (N_f \cdot N_p \cdot L_p) \quad (17)$$

where  $N_f$  is the number of forwarding neighbours,  $T_d$  is the amount of time for which nodes are producing data and  $L_{p_{beacon}}$  is the length of a beacon packet. This equation calculates the number of bits transmitted in the network by first calculating the total number of data packets transmitted, which is given in the last set of parentheses. The contribution from the beacon packets is calculated in the rest of the expression. Initially, the number of timeslots remaining after sleeping and after some have been used for data. Then the remaining number of slots are used to send beacon packets.

### 5.5. Energy consumption

In terms of energy analysis, only the power used to transmit and receive messages is considered. This is because the transceiver has the largest energy cost compared with that of the processor. The other factors such as the compiler used, which may make code more or less efficient, will affect the processors energy consumption, as well as other tasks that need to be run. There are also other energy costs attributed to things like the sensors and other peripherals, the mobility platform and the battery type, which are hardware specific and difficult to account for.

The energy consumption, EC, is characterised in terms of joules used per second per node:

$$EC = \left( \frac{V_{batt}}{R_b} \right) \cdot \left( \frac{(I_{tx} \cdot B_{tx}) + (I_{rx} \cdot B_{rx})}{n \cdot T_t} \right) \quad (18)$$

where  $V_{batt}$  is the voltage of the batteries,  $I_{tx}$  and  $I_{rx}$  are the current consumptions of the transceiver when transmitting and receiving respectively and  $B_{rx}$  is the total number of bits received.

Since RASER broadcasts packets to all neighbours,  $B_{rx}$  is given as

$$B_{rx} = B_{tx} \cdot N_n \quad (19)$$

This expression requires knowledge of the hardware but  $V_{batt}$ ,  $I_{tx}$  and  $I_{rx}$  can be substituted for temporary values based on potential hardware for comparison purposes.

## 6. Modelling and simulation results

RASER was modelled in the popular network simulator OPNET [46], which was used to run simulations of various network scenarios. For modelling purposes, the parameters used were designed to imitate the potential future application of autonomous UAV aided SAR, which is envisaged to consist of a medium sized swarm of drones and a manned team in a helicopter. The advantage of this is that the UAVs can be deployed immediately, without the delay of waiting for a crew to be assembled. This time saving can improve the chances of a target being found and potentially save lives. Additionally, the UAVs may report directly to the helicopter, which poses the additional challenge of a highly mobile sink. This particular application is well suited to RASER since the number of nodes deployed will be fixed for the lifetime of the mission and each UAV can be preprogrammed with an ID number in advance of their use. The UAVs may then be synchronised just as they are deployed, which will allow each of them to keep track of the current slot number. Subsequently, the slot numbers can be used by the nodes to indicate whose time slot it is, such that the sink will transmit in slot zero, then the node with ID number one will transmit in slot number one, node two will transmit in slot two and so



on. The slot length will also be known in advance, since the location information will have a fixed length and all nodes will have the same level of importance. Also, the location information is the only information that the helicopter team need, so as soon as one of the UAVs positively identifies a target, their position can be reported and the helicopter can be directed straight to the area in which they are needed.

Based on this, 25 nodes were placed in a square network area of 600 m by 600 m and a random waypoint mobility model was used, with a uniform distribution of speeds between 0 m/s and 25 m/s, and zero pause time. The maximum speed is close to both the cruising speed of a fixed wing UAV and the top speed of a rotary wing UAV. The random waypoint mobility model was used to imitate this varying topology of a cooperative search pattern, which may be executed by the UAVs.

The nodes were modelled on the Memsic IRIS motes [47], giving a transmission radius of 250 m and a bit rate of 250 kbps. The energy consumption parameters,  $V_{batt}$ ,  $I_{tx}$  and  $I_{rx}$ , were also modelled on the IRIS node, namely 3 v, 16.5 mA and 15.5 mA respectively. It should also be noted that RASer's queue length was limited to the number of nodes,  $n$ , in order to reduce the protocol's memory consumption and further increase the difficulty of the simulations.

A packet generation rate of 1 pk/s, for every node, was set to give a reasonable traffic level for a swarm detecting multiple targets within the search area. The size of the generated data was considered to be 32 bits, which is enough to report the node's position and some extra application specific data.

RASer is compared with four other routing protocols; PHASer, MACRO, AODV and OLSR. The popular ZigBee standard [48], is based on AODV and also one of the most commonly used protocols in sensor networks. For this reason AODV is used as a performance baseline of general sensor networks. Since AODV is reactive, the proactive protocol, OLSR, is included in our results as a comparison for completeness. MACRO is a recent high performance routing protocol, which is designed specifically for MWSNs. MACRO is used as a representation of the current developments in MWSN routing protocols. PHASer is also included as a comparison as it is a recently published MWSN routing protocol and it shares some common founding concepts with RASer, though the resulting protocols are quite different. In addition, analytical results for a flooding protocol using the GTDMA MAC layer are given. This is done in order to assess the impact of blind forwarding in comparison to a flooding protocol. This will also demonstrate the strength of GTDMA and how using a collision free MAC has a heavy influence on a protocol's reliability. In terms of the parameters used in the analytical framework from Section 5, the flooding protocol increases the number of forwarding neighbours to be  $n-1$  and also increases the packet arrival rate to be the product of the packet creation rate and the expected number of node whose data will be received.

Both AODV and OLSR were implemented with a hello interval of one second, whereas MACRO used a hello interval of five seconds since it is able to cope with the mobility of the nodes better than the other two. MACRO's active entry timeout was also set to five seconds. In order to improve the results from AODV a little further, local route repair was enabled and the route timeout was kept down to one second. The physical layer, mobility and traffic generation parameters are the same for all protocols.

The analytical results are modelled on a FCFS basis rather than with the more complex use of priorities, so the results shown are expected to characterise the general performance of the protocol rather than either normal or supersede mode.

Results are gathered for PDR, average end-to-end delay, overhead, throughput and energy consumption. All of the metrics are as defined in Section 5.

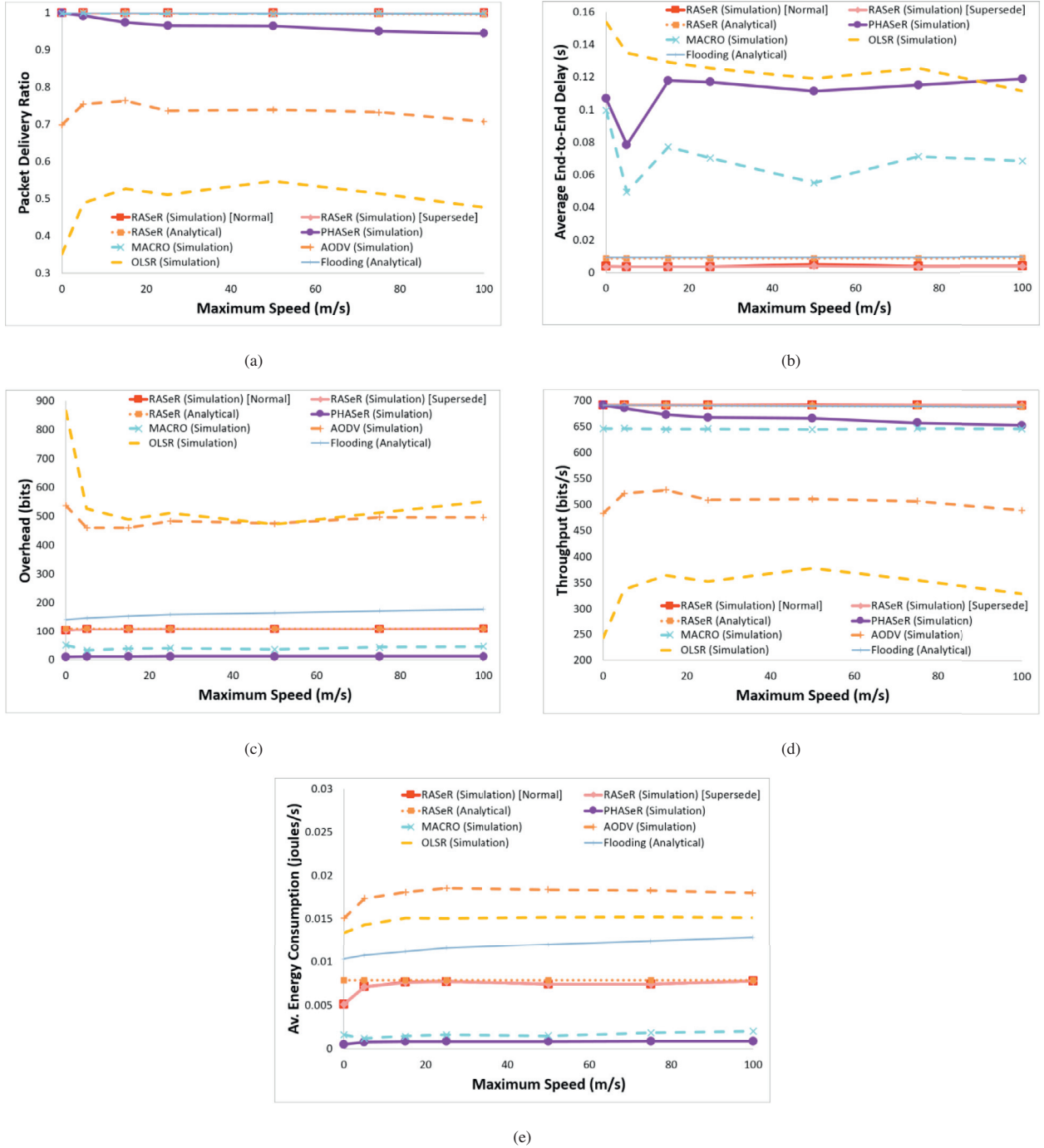
Parameters are varied around the base values given above to test the protocol's performance with varying levels of mobility, scalability and traffic.

### 6.1. Mobility

In this set of simulations the maximum speed was varied to evaluate how RASer performs under different levels of mobility. The nodes' speeds are randomly chosen by the mobility model from a range of 0 m/s to [0, 5, 15, 25, 50, 75, 100] m/s. The rest of the parameters remained constant as described above. Fig. 6 gives the mobility results for all five metrics. The PDR for both RASer and MACRO are near perfect, with RASer improving on MACRO by only about 0.04%. Both protocols show an improvement over PHASer and vast superiority over AODV and OLSR. Additionally, the analytical results give a close approximation to the simulation results. The flooding analytical results give marginally worse PDR than RASer. In terms of delay, AODV gives by far the worst performance and has been removed from the figure such that the scale can be increased to better display the other results. OLSR shows better delay, but this is mainly due to a high level of packet loss. RASer's delay is consistently less than 5 ms, which is a large improvement over PHASer and MACRO, whose lowest delay is over 49 ms. This is also reinforced by the throughput results and again the analytical expression is a good approximation of the simulation. The results for energy and overhead clearly show PHASer to be the least power hungry, followed closely by MACRO. RASer has a good level of energy consumption and is an improvement over both OLSR and AODV. The analytical results are close for overhead, but show a slight overestimation for energy, especially at lower speeds. The flooding results are consistently good, although marginally inferior to RASer, which demonstrates the advantage of using a collision free MAC layer.

### 6.2. Scalability

Fig. 7 gives the results for scalability, in which the number of nodes is varied between [15, 25, 50, 75, 100] nodes. The size of the square network was also varied to keep the scenarios realistic. The length of each side was set to [400, 600, 1000, 1200, 1500] m for the number of nodes accordingly. The packet generation rate and maximum speed were kept consistent at 1 pk/s and 25 m/s respectively. Over all scales RASer has a very high PDR, whereas both PHASer and MACRO shows a significant deterioration at 50 nodes and higher. As with the mobility results, RASer has a consistently low delay, with a slight increase at 100 nodes. These results highlight the difference between normal mode and supersede mode; where normal mode maintains a high PDR in the 100 node scenario, supersede mode shows a small drop. Whereas the delay results show supersede mode to maintain a low delay and normal mode to have a small increase. This comes from supersede mode dropping old packets in favour of lower delay times, compared with normal mode trying to deliver every packet at the cost of an increase in delay. A more extreme version of this is in the OLSR results, which shows a very low delay but at the cost of between 18% and 85% lost packets. Whilst not so extreme, PHASer also manages to maintain a low delay, overhead and energy consumption, at the expense of the PDR and throughput. Contrastingly, AODV has a large increase in delay, overhead and energy consumption as the number of nodes increases, as does MACRO, although to a lesser extent. For all protocols the throughput results increase as more nodes are introduced to the network, with RASer giving the best performance overall. The overhead and energy results illustrate how RASer becomes more energy efficient as the number of nodes increases, which is due to the fact the slot time is the same for each scenario. In this way, the number of slots over



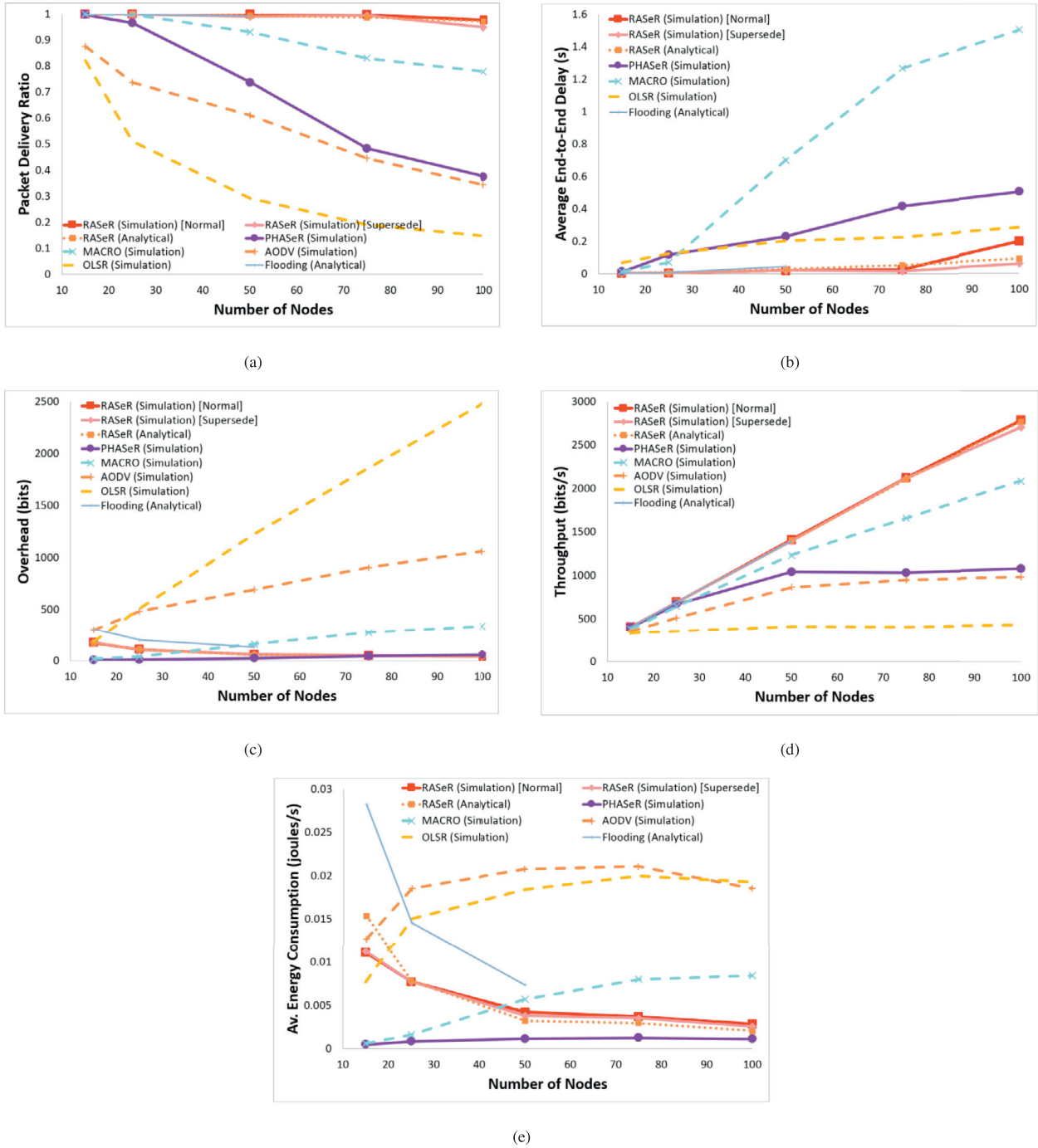
**Fig. 6.** Performance results of RASeR in comparison with PHASeR, MACRO, AODV and OLSR, over varying maximum speeds: (a) PDR, (b) average end-to-end delay (AODV's average result was 1.23 s and was removed to improve the scale), (c) overhead, (d) throughput and (e) average energy consumption. RASeR simulation results are given for both normal and supersede modes. Analytical results are also shown for RASeR and a basic flooding protocol using the GTDMA MAC layer. These results have a maximum 95% confidence interval of 1.59%.

a time period remains the same, only allowing a fixed number of transmissions. So even if there are lots of nodes, there are still only a limited number of time slots within which they may transmit. As such, the increased number of nodes causes the average energy to decrease. As with the mobility results, the flooding shows slightly worse performance than RASeR, though this becomes more significant when considering the overhead and energy metrics. It should also be noted that the flooding curve doesn't continue past 50 nodes, this is because the system becomes unstable. Stability requires that the utilisation,  $\rho = \lambda T_s$ , is less than one. However, with

75 and 100 nodes  $\rho > 1$ , and as such the nodes will be receiving packets faster than they can service them causing the network to experience high levels of delay and packet loss.

### 6.3. Traffic

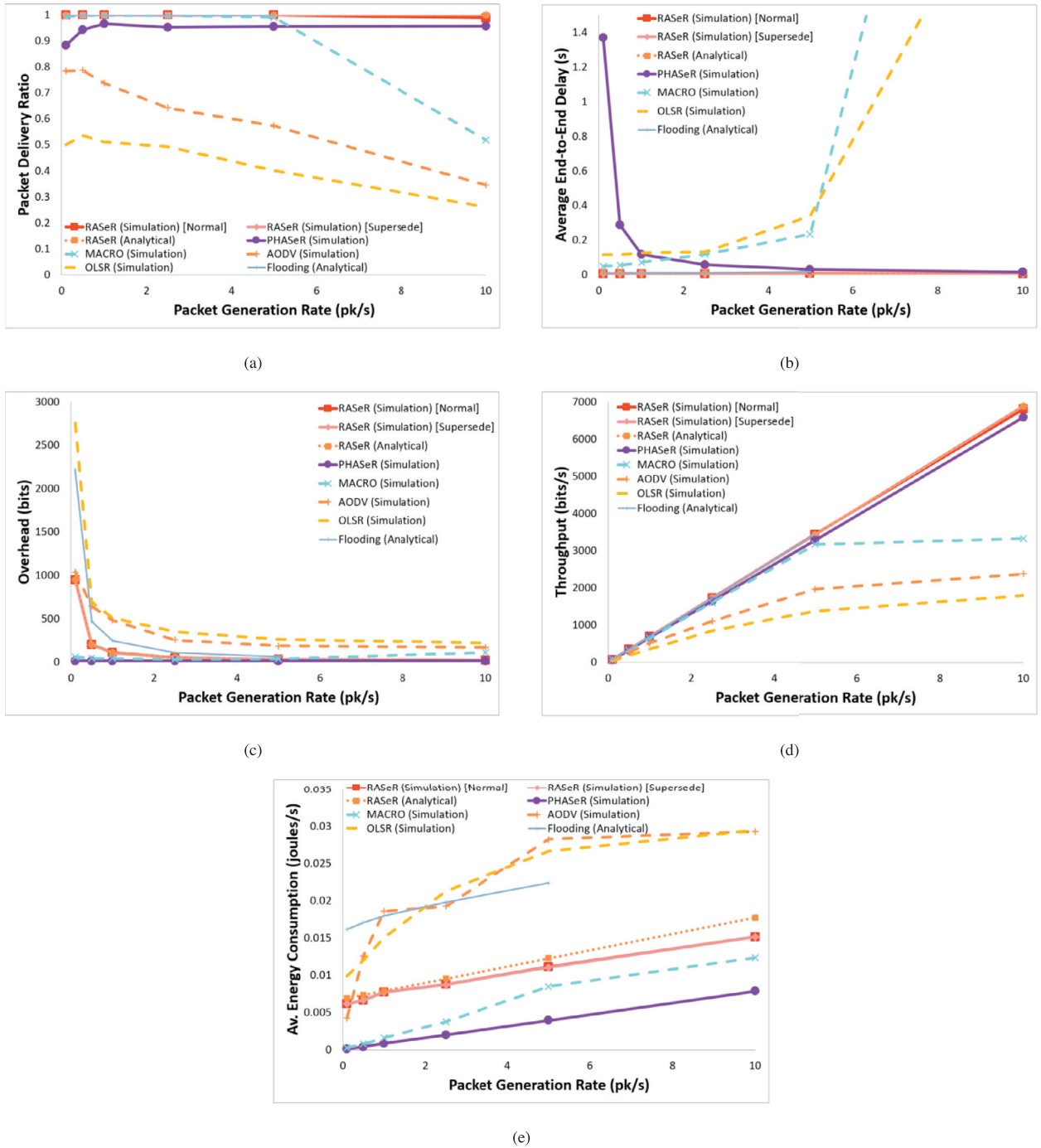
In order to evaluate RASeR's performance under varying traffic loads the packet arrival rate per node was varied. Fig. 8 gives results for [0.1, 0.5, 1, 2.5, 5, 10] pk/s. Again, all other parameters remained static with 25 nodes in a 600 m by 600 m area



**Fig. 7.** Performance results of RASeR in comparison with PHASeR, MACRO, AODV and OLSR, over varying numbers of nodes: (a) PDR, (b) average end-to-end delay (AODV's average result was 5.13 s and was removed to improve the scale), (c) overhead, (d) throughput and (e) average energy consumption. RASeR simulation results are given for both normal and supersede modes. Analytical results are also shown for RASeR and a basic flooding protocol using the GTDMA MAC layer. These results have a maximum 95% confidence interval of 1.36%.

and a maximum speed of 25 m/s. Both RASeR and MACRO have near perfect PDR over all packet generation rates, until MACRO drops to almost 50% at 10 pk/s. PHASeR gives a slightly worse performance than both RASeR and MACRO, but stays relatively consistent over the entire range. AODV and OLSR show significantly worse performance, which deteriorates further at higher packet generation levels. Again, in terms of delay, RASeR shows very good results with minimal variation and greatly outperforms MACRO, AODV and OLSR, especially at high packet generation rates.

MACRO and OLSR begin well but have a large increase at 10pk/s and as such their final points are not shown on the figure, such that the other results can be more closely inspected. The throughput results show RASeR and PHASeR to have a steady increase as more packets are generated. MACRO's throughput increases to begin with but then plateaus after 5 pk/s, which indicates its upper bound in this scenario. Results for overhead highlight how RASeR's overhead decreases as the packet generation rate increases, which is due to more time slots being filled with data, improving its efficiency. This highlights how, when used with low traffic levels, the

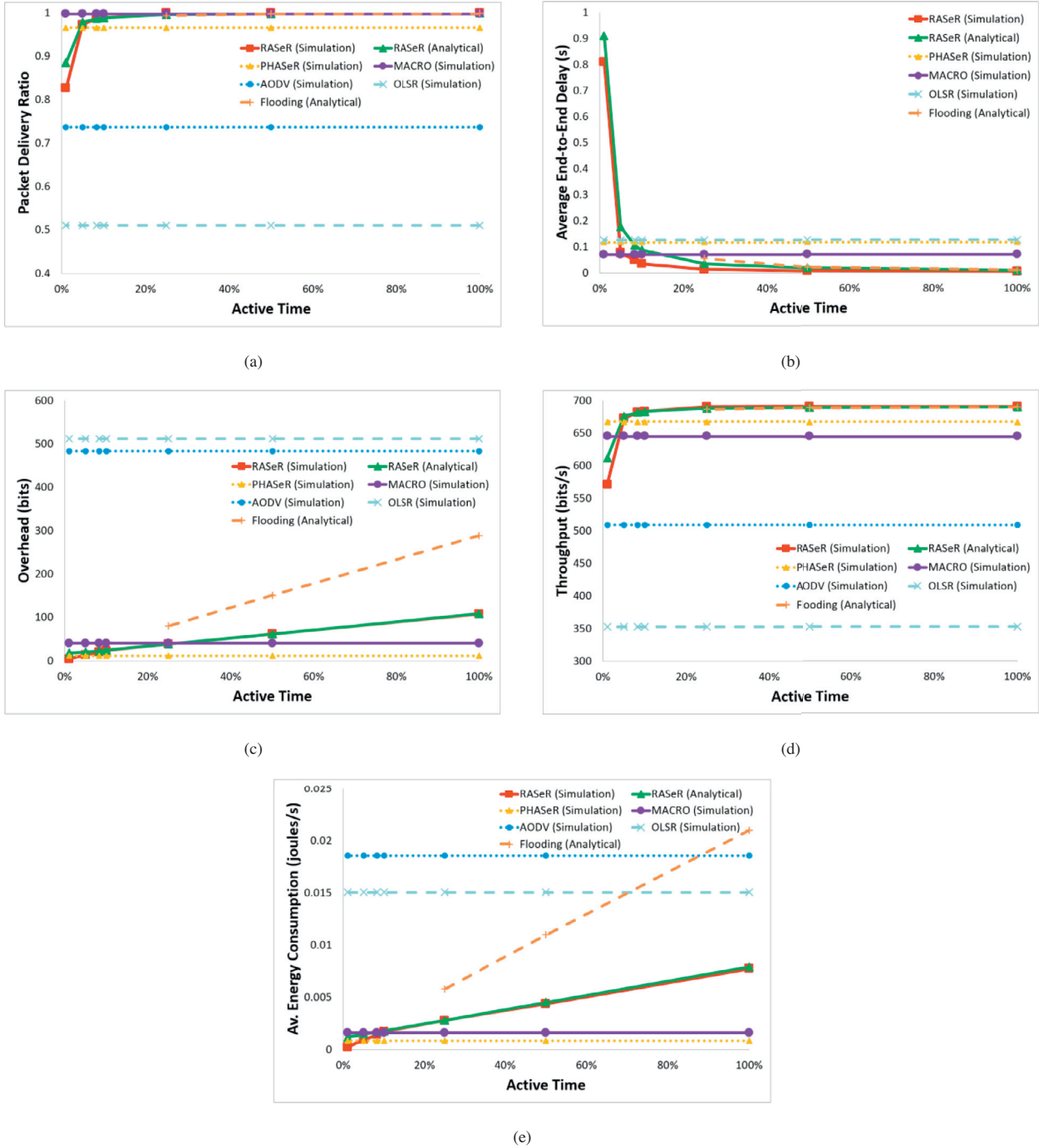


**Fig. 8.** Performance results of RASeR in comparison with PHASeR, MACRO, AODV and OLSR, over varying packet generation rates: (a) PDR, (b) average end-to-end delay (AODV's average result was 18.88 s, MACRO's final value was 5.02 s, OLSR's final value was 2.53 s. These points were removed to improve the scale), (c) overhead, (d) throughput and (e) average energy consumption. RASeR simulation results are given for both normal and supersede modes. Analytical results are also shown for RASeR and a basic flooding protocol using the GTDMA MAC layer. These results have a maximum 95% confidence interval of 4.22%, which occurs in the lowest packet generation rates, which reduces the sample size.

overhead is comparatively large. However, when the data generation rate is higher, the overhead becomes proportionally smaller. This also occurs when the payload size is increased, since there are more data bits being successfully delivered in comparison to the amount of overhead being produced. The overhead of AODV and OLSR decreases due to their increased packet loss. Contrastingly, MACRO only shows a very slight increase in overhead, which is also reflected in its energy results. Whereas, PHASeR yields a very steady overhead and its energy consumption increases at a lesser

rate. Both AODV and OLSR have very high energy consumption considering their low packet loss. Whereas, RASeR shows a reasonable increase in energy consumption given that there is more data in the network. Also, the analytical results are very close to the simulated results for all metrics, except average energy consumption, which is slightly overestimated. In the same way as with the scalability results, the flooding protocol gives worse results than RASeR, especially in overhead and energy consumption and again the protocol becomes unstable after 5 pk/s.



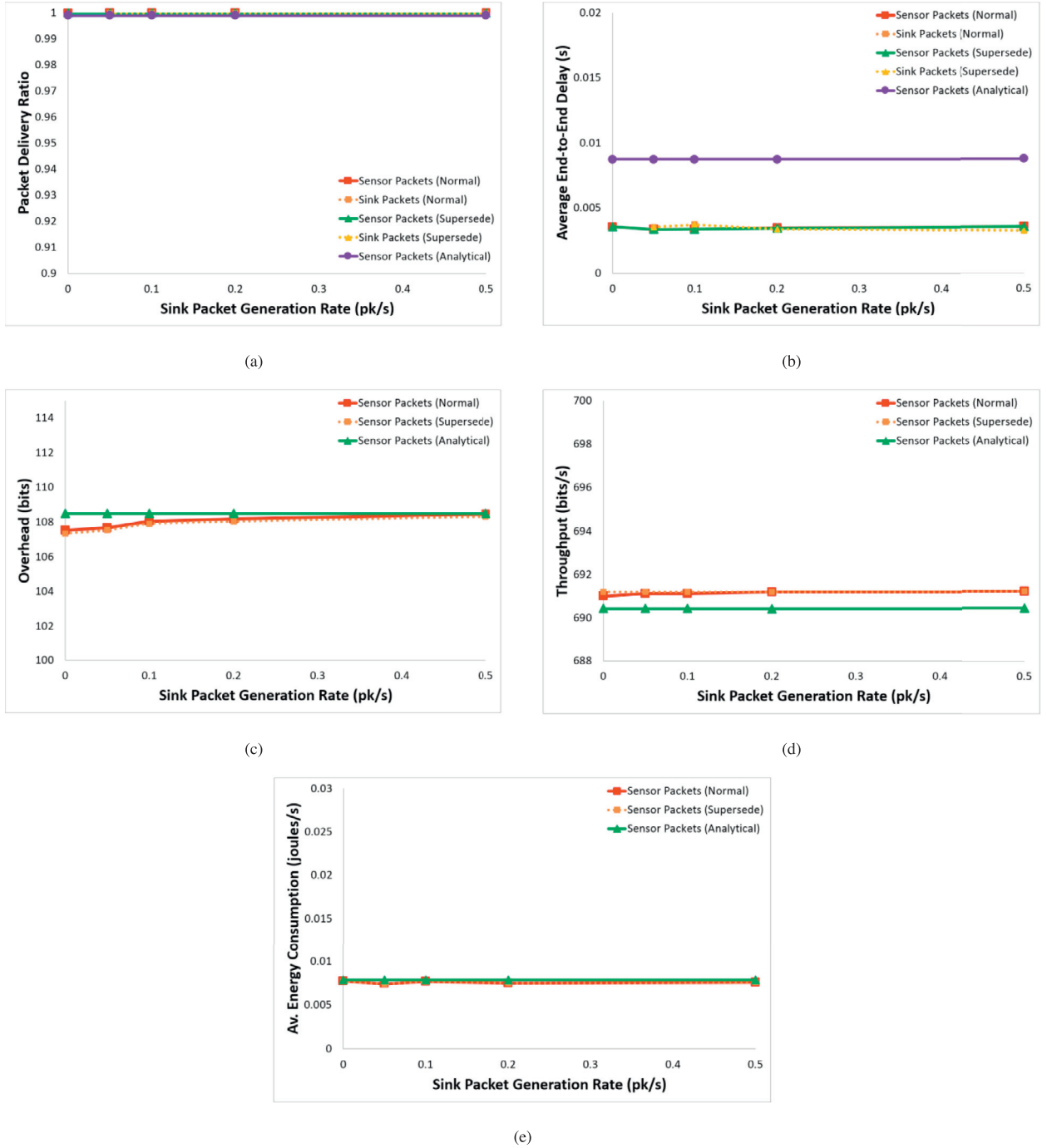


**Fig. 9.** Performance results of RASeR over varying amounts of active time: (a) PDR, (b) average end-to-end delay (AODV's average result was 1.07 s and was removed to improve the scale), (c) overhead, (d) throughput and (e) average energy consumption. With fixed PHASeR, MACRO, AODV and OLSR levels shown for comparison. Analytical results are also shown for RASeR and a basic flooding protocol using the GTDMA MAC layer. These results have a maximum 95% confidence interval of 1.32%.

#### 6.4. Energy saving

The results given in Fig. 9 illustrate the effect of using sleep cycles in RASeR. The results are given in terms of the fraction of time in a round, for which the network is active as opposed to being asleep. Data was gathered for [100, 50, 25, 10, 8.33, 5, 1] active time. All other parameters remained constant with 25 nodes in a 600 m by 600 m network, each producing packets at a rate of 1 pk/s and a maximum speed of 25 m/s. Results for PHASeR, MACRO, AODV and OLSR have been plotted as straight lines giving performance indicators for comparison purposes; even though

changing the active time in RASeR will have no effect on the other three protocols. With very low active time the results show that RASeR's overall performance decreases. However, with 25% active time and above, RASeR's PDR, end-to-end delay and throughput remain relatively unchanged. The overhead and energy results highlight the significant saving given by the added sleep cycles. Over all metrics, RASeR has good performance, even with very low energy consumption. It consistently outperforms AODV and OLSR. With the exception of the delay metric, which shows OLSR dropping nearly half of all its packets in order to route data faster. When RASeR is compared with MACRO there is a clear trade-off between



**Fig. 10.** Performance results of RASer over varying sink packet generation rates: (a) PDR, (b) average end-to-end delay, (c) overhead, (d) throughput and (e) average energy consumption. RASer simulation results are given for both normal and supersede modes.

the energy and overhead metrics, and the PDR, delay and throughput metrics; with slightly higher energy consumption and overhead, RASer achieves exceptional performance in PDR, delay and throughput, superior to that of MACRO. Whereas, when the active time is low, the energy consumption is better than MACRO, but the PDR, delay and throughput are worse. The cross over point appears to be at around 8.33% active time, where supersede mode has better performance over all metrics in comparison to MACRO. When compared with PHASer, with a high active time, RASer shows better performance in PDR, average delay and throughput. However, at 5%, both protocols show comparable energy and overhead results, but RASer still shows an improved PDR, delay and throughput. The

analytical flooding results show that initially the system is unstable, but after 25% they yield good performance in everything except overhead and energy, which give relatively high values.

### 6.5. Reverse flooding

These results are given to evaluate whether a high level of packets broadcast by the sink, will affect the data packets being transmitted by the sensors. Fig. 10 gives results over sink packet arrival rates of [0, 0.05, 0.1, 0.2, 0.5] pk/s, since one network-wide command every two seconds is anticipated to be sufficient for most applications. Again, all other parameters were kept static with 25

nodes, a maximum speed of 25 m/s. The network area was 600 m by 600 m and each sensor had a packet generation rate of 1 pk/s. The results show that, compared with zero sink packets per second, the results remain fairly constant, with no severe degradation on any of the metrics. Overhead only shows an increase of approximately 1 bit at the highest sink packet generation rate. Additionally, PDR and average end-to-end delay results have also been gathered for the sink packets themselves. The PDR is the ratio of packets created to packets received by all nodes, so a perfect PDR would indicate that the sink packet has been received successfully by every node. The end-to-end delay is calculated as the average time the packet takes to reach the node after it has been generated by the sink. These results indicate that the sink packets are delivered with near 100% PDR and with an average delay of less than 4 ms, this trend shows no signs of changing and is expected to continue for higher sink generation rates.

## 6.6. Discussion

The results demonstrate RASer's high level of performance across all metrics and show its ability to cope with a large variety of scenarios.

The proposed is shown to perform significantly better than both OLSR and AODV in all scenarios. The results also show how RASer outperforms PHASer in PDR, delay and throughput in every scenario. However, PHASer's packet loss and low end-to-end delay allow it to maintain a reduced level of overhead and energy consumption. In comparison to MACRO, RASer generally has significantly better end-to-end delay, which is a key concern for future applications. In most cases RASer also shows a slightly improved PDR. The combination of low delay and high PDR gives RASer much higher throughput than MACRO, over all scenarios. However, although RASer's energy consumption is low, in scenarios with 25 nodes or less, MACRO tends to consume less energy. Additionally, the energy saving results have shown RASer's ability to still maintain a high level of performance, even in low energy cases. As such, with the use of RASer's energy saving mechanism, the trade-off between performance and energy consumption can be adjusted to fit the needs of the application.

The analytical expressions were also compared to the simulation results and have been found to characterise the trends of the protocol well. In most cases they also give close estimates to the simulated values. It is also interesting to see how the flooding protocol on top of the GTDMA MAC gives good levels of PDR, delay and throughput, highlighting the advantage of a collision free MAC. However, this comes at the cost of higher levels of overhead and energy consumption, which can cause the network to become unstable. This suggests that the use of blind forwarding reduces the overhead and improves on the flooding protocol, enabling it to cope with higher numbers of nodes and higher amounts of traffic.

## 7. Conclusion

This paper has presented RASer, a novel routing protocol designed for MWSNs, which has been shown to give a high level of performance in very demanding scenarios. Its unique use of a GTDMA MAC layer facilitates the maintaining of a simple hop count gradient at each node. This enables the use of blind forwarding to route data towards the sink. Reliable packet delivery is achieved through the protocol's inherent use of route diversity and resilience to link breakages.

RASer is suited for many uses in MWSNs and is further enhanced for other application requirements with the addition of a supersede mode, which gives precedence to the latest data rather than trying to deliver every packet. Also, reverse flooding is included, which is a simple mechanism to allow the sink to commu-

nicate with the sensor nodes. Another important part of the protocol is the energy saving mechanism, which reduces power consumption by introducing sleep cycles.

The simulations have verified the protocols suitability for time critical applications by demonstrating its ability to consistently deliver data with low latency. The supersede mode, reverse flooding and energy saving aspects of the protocol were also tested and found to be valuable additions.

The protocol was compared with OLSR and AODV, which is the basis of the commonly used ZigBee protocol. The results found that RASer was superior in all scenarios, for all metrics. The MWSN protocol PHASer was also simulated and even though some of the core concepts are similar to RASer, the differences in the two protocols can clearly be seen in the vastly different levels of performance. PHASer sacrifices PDR, delay and throughput for consistently low overhead and energy consumption. Whereas, RASer achieves very high levels of PDR and throughput with minimal latency, whilst also maintaining a relatively low level of energy consumption. Additionally, MACRO is a high performance routing protocol, also designed for MWSNs. In comparison, RASer was shown to have marginally improved PDR, better throughput and superior end-to-end delay. However, in networks with low numbers of nodes, even though RASer's power consumption is still low, MACRO consumed less energy and produced less overhead.

Analytical expressions were given to characterise the protocol's performance and subsequently the simulated results have shown that RASer can cope with very high mobility levels, with near perfect PDR and low end-to-end delay times. The same is true under varying traffic loads, which highlights RASer's adaptability to different scenarios. The protocol is also scalable to large numbers of nodes. Additionally, future work will look at implementing RASer on a testbed to further verify its capabilities and suitability for various applications.

## Acknowledgement

This work is partly supported by the UK Engineering and Physical Sciences Research Council (EPSRC) grant number EP/K503198/1. All data created may be accessed upon request.

## References

- [1] I.F. Akyildiz, Weilian Su, Y. Sankarasubramanian, E. Cayirci, A survey on sensor networks, *IEEE Commun. Mag.* 40 (8) (Aug. 2002) 102–114.
- [2] M. Conti, S. Giordano, Mobile Ad Hoc networking: milestones, challenges and new research directions, *IEEE Commun. Mag.* 52 (1) (Jan. 2014) 85–96.
- [3] S. Ehsan, et al., Design and analysis of delay-tolerant sensor networks for monitoring and tracking free-roaming animals, *IEEE Trans. Wireless Commun.* 11 (3) (2012) 1220–1227.
- [4] B. White, et al., Contaminant cloud boundary monitoring using network of UAV sensors, *IEEE Sensors J.* 8 (10) (2008) 1681–1692.
- [5] X. Li, et al., Performance evaluation of vehicle-based mobile sensor networks for traffic monitoring, *IEEE Trans. Veh. Technol.* 58 (4) (2009) 1647–1653.
- [6] K. Lorincz, et al., Sensor networks for emergency response: challenges and opportunities, *IEEE Pervasive Comput.* 3 (4) (2004) 16–23.
- [7] H. Yan, H. Huo, Y. Xu, M. Gidlund, Wireless sensor network based E-health system – implementation and experimental results, *IEEE Trans. Consum. Electron.* 56 (4) (2010) 2288–2295.
- [8] S. Waharte, N. Trigoni, Supporting search and rescue operations with UAVs, in: *Proceedings of Int'l Conference on Emerging Security Technologies (EST '10)*, Sept. 2010, pp. 142–147.
- [9] S. Bohacek, Performance improvements provided by route diversity in multi-hop wireless networks, *IEEE Trans. Mobile Comput.* 7 (3) (Mar. 2008) 372–384.
- [10] J. Al-Karaki, A. Kamal, Routing techniques in wireless sensor networks: a survey, *IEEE Wireless Commun.* 11 (6) (2004) 6–28.
- [11] C.E. Perkins, E.M. Royer, Ad-hoc on-demand distance vector routing, in: *Proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, 1999, pp. 90–100.
- [12] D.B. Johnson, D.A. Maltz, Dynamic source routing in Ad Hoc wireless networks, in: T. Imielinski, H. Korth (Eds.), *Mobile Computing*, Kluwer Academic Publishers, 1996, pp. 153–181.
- [13] S. Biswas, R. Morris, ExOR: opportunistic multi-hop routing for wireless networks, *ACM SIGCOMM Comput. Commun. Rev.* 35 (4) (Oct. 2005) 133–144.

- [14] S. Ren, et al., An improved wireless sensor networks routing protocol based on AODV, in: *Proceedings of IEEE 12th Int'l Conference on Computer and Information Technology (CIT '12)*, Oct. 2012, pp. 742–746.
- [15] H. Soliman, M. Alotaibi, An efficient routing approach over mobile wireless Ad-Hoc sensor networks, in: *Proceedings of 6th IEEE Consumer Communications and Networking Conference (CCNC '09)*, Jan. 2009, pp. 1–5.
- [16] G. Huo, X. Wang, An opportunistic routing for mobile wireless sensor networks based on RSSI, in: *Proceedings of 4th Int'l Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '08)*, Oct. 2008, pp. 1–4.
- [17] Y. Han, Z. Lin, A geographically opportunistic routing protocol used in mobile wireless sensor networks, in: *Proceedings of 9th IEEE Int'l Conference on Networking, Sensing and Control (ICNSC)*, Apr. 2012, pp. 216–221.
- [18] S. Kwangcheol, K. Kim, S. Kim, ADSR: angle-based multi-hop routing strategy for mobile wireless sensor networks, in: *Proceedings of IEEE Asia-Pacific Services Computing Conference (APSCC)*, Dec. 2011, pp. 373–376.
- [19] X. Huang, H. Zhai, Y. Fang, Robust cooperative routing protocol in mobile wireless sensor networks, *IEEE Trans. Wireless Commun.* 7 (12) (Dec. 2008) 5278–5285.
- [20] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy efficient communication protocol for wireless micro sensor networks, in: *Proceedings of 33rd Hawaii Int'l Conference on System Sciences (HICSS '00)*, 2000, p. 8020.
- [21] D. Kim, Y. Chung, Self-organization routing protocol supporting mobile nodes for wireless sensor network, in: *Proceedings of 1st Int'l Multi-Symposiums on Computer and Computational Sciences (IMSCS '06)*, June 2006, pp. 622–626.
- [22] G.S. Kumar, M.V. Vinu, P.G. Athithan, K.P. Jacob, Routing protocol enhancement for handling node mobility in wireless sensor networks, in: *Proceedings of IEEE Region 10 Conference (TENCON)*, 2008, pp. 1–6.
- [23] S.A.B. Awwad, C.K. Ng, N.K. Noordin, M.F.A. Rasid, Cluster based routing protocol for mobile nodes in wireless sensor network, in: *Proceedings of Int'l Symposium on Collaborative Technologies and Systems (CTS '09)*, 2009, pp. 233–241.
- [24] R.U. Anitha, P. Kamalakkannan, Enhanced cluster based routing protocol for mobile nodes in wireless sensor network, in: *Proc. Int'l Conf. Pattern Recognition, Information and Medical Engineering (PRIME)*, 2013, pp. 187–193.
- [25] S. Deng, J. Li, L. Shen, Mobility-based clustering protocol for wireless sensor networks with mobile nodes, *IET Wireless Sensor Syst.* 1 (1) (Mar. 2011) 39–47.
- [26] U. Ahmed, F.B. Hussain, Energy efficient routing protocol for zone based mobile sensor networks, in: *Proceedings of 7th Int'l Wireless Communications and Mobile Computing Conference (IWCMC)*, 2011, pp. 1081–1086.
- [27] L. Karim, N. Nasser, Reliable location-aware routing protocol for mobile wireless sensor network, *IET Commun.* 6 (14) (Sept. 2012) 2149–2158.
- [28] J. Kulik, W. Heinzelman, H. Balakrishnan, Negotiation-based protocols for disseminating information in wireless sensor networks, *Wireless Netw.* 8 (2/3) (Mar. 2002) 169–185.
- [29] C. Intanagonwiwat, et al., Directed diffusion for wireless sensor networking, *IEEE/ACM Trans. Netw.* 11 (1) (Feb. 2003) 2–16.
- [30] A. Aronsky, A. Segall, A multipath routing algorithm for mobile wireless sensor networks, in: *Proceedings of 3rd Joint IFIP Wireless and Mobile Networking Conference*, Oct. 2010, pp. 1–6.
- [31] S. Cakici, et al., A novel cross-layer routing protocol for increasing packet transfer reliability in mobile sensor networks, *Springer Wireless Personal Commun. J.* 77 (3) (Aug. 2014) 2235–2254.
- [32] C. Shurgers, M.B. Srivastava, Energy efficient routing in wireless sensor networks, in: *Proceedings of IEEE Military Communications Conference (MILCOM '01)*, 2001, pp. 357–361.
- [33] B. Karp, H. Kung, GPSR: greedy perimeter stateless routing for wireless networks, in: *Proceedings of 6th Annual Int'l Conference on Mobile Computing and Networking (MobiCom '00)*, 2000, pp. 243–254.
- [34] L. Zou, M. Lu, Z. Xiong, A distributed algorithm for the dead end problem of location based routing in sensor networks, *IEEE Trans. Veh. Technol.* 54 (4) (July 2005) 1509–1522.
- [35] L. Zou, M. Lu, Z. Xiong, PAGER-M: a novel location based routing protocol for mobile sensor networks, in: *Proceedings of Broadwise*, Oct. 2004.
- [36] R. Kouah, M. Aissani, S. Moussaoui, Mobility-based greedy forwarding mechanism for wireless sensor networks, in: *Proceedings of 9th Int'l Conference on Networking and Services (ICNS '13)*, Mar. 2013, pp. 140–145.
- [37] R. Jurdak, et al., Directed broadcast with overhearing for sensor networks, *ACM Trans. Sensor Netw.* 6 (1) (2009) 3:1–3:35.
- [38] H. Liu, et al., Opportunistic routing for wireless Ad Hoc and sensor networks: present and future directions, *IEEE Commun. Mag.* 47 (12) (Dec. 2009) 103–109.
- [39] T. Hayes, F.H. Ali, Proactive highly ambulatory sensor routing (PHASer) protocol for mobile wireless sensor networks, *Elsevier Pervasive Mobile Comput.* 21 (Aug. 2015) 47–61.
- [40] F. Sivrikaya, B. Yener, Time synchronization in sensor networks: a survey, *IEEE Netw.* 18 (4) (2004) 45–50.
- [41] J. Elson, L. Girod, D. Estrin, Fine-grained time synchronization using reference broadcasts, in: *Proceedings of 5th Symposium on Operating Systems Design and Implementation*, Dec. 2002, pp. 147–163.
- [42] W. Zhu, TDMA frame synchronization of mobile stations using radio clock signal for short range communications, in: *Proceedings of 44th IEEE Vehicular Technology Conference*, June 1994, pp. 1878–1882.
- [43] S. Ganeriwal, et al., Timing-sync protocol for sensor networks, in: *Proceedings of 1st Int'l Conference on Embedded Networked Sensor Systems (SenSys '03)*, 2003, pp. 138–149.
- [44] P. Samar, S.B. Wicker, On the behavior of communication links of a node in a multi-hop mobile environment, in: *Proceedings of 5th ACM Int'l Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '04)*, 2004, pp. 145–156.
- [45] P.A. Contla, M. Stojmenovic, Estimating hop counts in position based routing schemes for Ad Hoc networks, *Telecommun. Syst.* 22 (1–4) (2003) 109–118.
- [46] OPNET Technologies Inc, OPNET 2013; [www.opnet.com](http://www.opnet.com).
- [47] Memsic Inc, IRIS 2014; [http://www.memsic.com/userfiles/files/Datasheets/WSN/IRIS\\_Datasheet.pdf](http://www.memsic.com/userfiles/files/Datasheets/WSN/IRIS_Datasheet.pdf).
- [48] J. Li, et al., Study on ZigBee network architecture and routing algorithm, in: *Proceedings of 2nd Int'l Conference on Signal Processing Systems (ICSPS)*, July 2010, pp. 389–393.





**Tom Hayes** received the B.Sc. in Audio Systems Engineering from the University of the West of England in 2011 and M.Sc. in Embedded Digital Systems from the University of Sussex in 2012. He has been pursuing his Ph.D. in the Communications Research Group at the University of Sussex since 2012, for which he is partly funded by EPSRC. Research interests include mobile wireless sensor networks, routing protocols and cooperative communications.



**Falah H. Ali** holds B.Sc. in Electrical and Electronics Engineering and M.Sc. in Electronic Systems from Cardiff University in 1984 and 1986, respectively, and Ph.D. in the area of multiple access communications from the University of Warwick in 1992. During 1992–1994 he was a postdoctoral research associate at the University of Lancaster. In 1994 he joined the University of Sussex as a Lecturer in Electronics Engineering and in 2000 appointed as a Senior Lecturer. Currently he is a Reader in Digital Communications and Director of Communications Research Group at the University of Sussex. He leads an active research team undertaking high quality research on a wide range of projects. His research interests include mobile communications and wireless sensor networks. He co-authored over 100 research papers, supervised many Ph.D. research students, and served on a number of international conference committees. He is a Fellow of IET, Senior Member of IEEE, and Chartered Engineer.